# A FOCUSED WEB CRAWLER

**Dinesh Kannan**

**kannan.d@husky.neu.edu**

## SUMMARY:

The goal of this web crawler is to specifically search for a keyphrase in webpages starting from a given seed page. This crawler determines the relevance of a page by simply checking if the keyphrase is present in the visible parts of a webpage. Thereby this crawler crawls only the links of relevant webpages. The crawling stops when the depth of crawling has reached a depth of 5 webpages from the seed page or has visited 1000 unique relevant pages, whichever occurs first. This crawler is implemented in Python and requires the following modules,

- urllib
- bs4
- urlparse
- robotparser
- time
- socket

## CRAWLER SPECIFICS:

This crawler takes in two arguments, the seedpage and the keyphrase

Complying with the problem statement, this crawler crawls only,

- English-Wikipedia pages
- Non-administrative pages (URLs without ':' in URL path)

Only the contents of a webpage are examined for relevance irrespective of whether the links contain the keyphrase.

## DESIGN:

## IMPLEMENTATION:

This crawler has been split into the following modules, each serving a specific functionality.

- crawl
- filter
- urlValidity
- checkRobots

- searchKeyPhrase

1. **crawl :** This is the main module of the crawl functionality. The crawler calls this module immediately after the start of the program. This module contains various sub-modules which collectively serve the role of a focused web crawler. This module calls the filter module for filtering out the URLs, which is the first step in determining relevance. If a page is found relevant and crawlable, this module then retrieves all the outlinks from this page and places them in the URLRequest Queue.

2. **filter:** This module is for filtering out the URLs based on conditions in the problem statement**.** This module sequentially checks the following things,
    - If the URL has already been blocked or visited
    - If the URL refers to an admin page (':' in URL path)
    - If the URL refers to the main page  (Main_Page)
    - If the URL refers to English-Wikipeida pages only
    - If the URL is valid by calling urlValidity module
    - If the URL is crawlable by checking robots.txt using checkRobots module
    - If the page contains the keyphrase by calling searchKeyPhrase module

    This module rejects the URL if any one of the above checks fail.

3. **urlValidity:** This module is called by the filter module to check if the URL is valid. i.e. refers to an active and actual webpage. This checks if the pages can be opened without any network or connection issues. For this, pages which returns RESPONSE CODE :200 are only considered

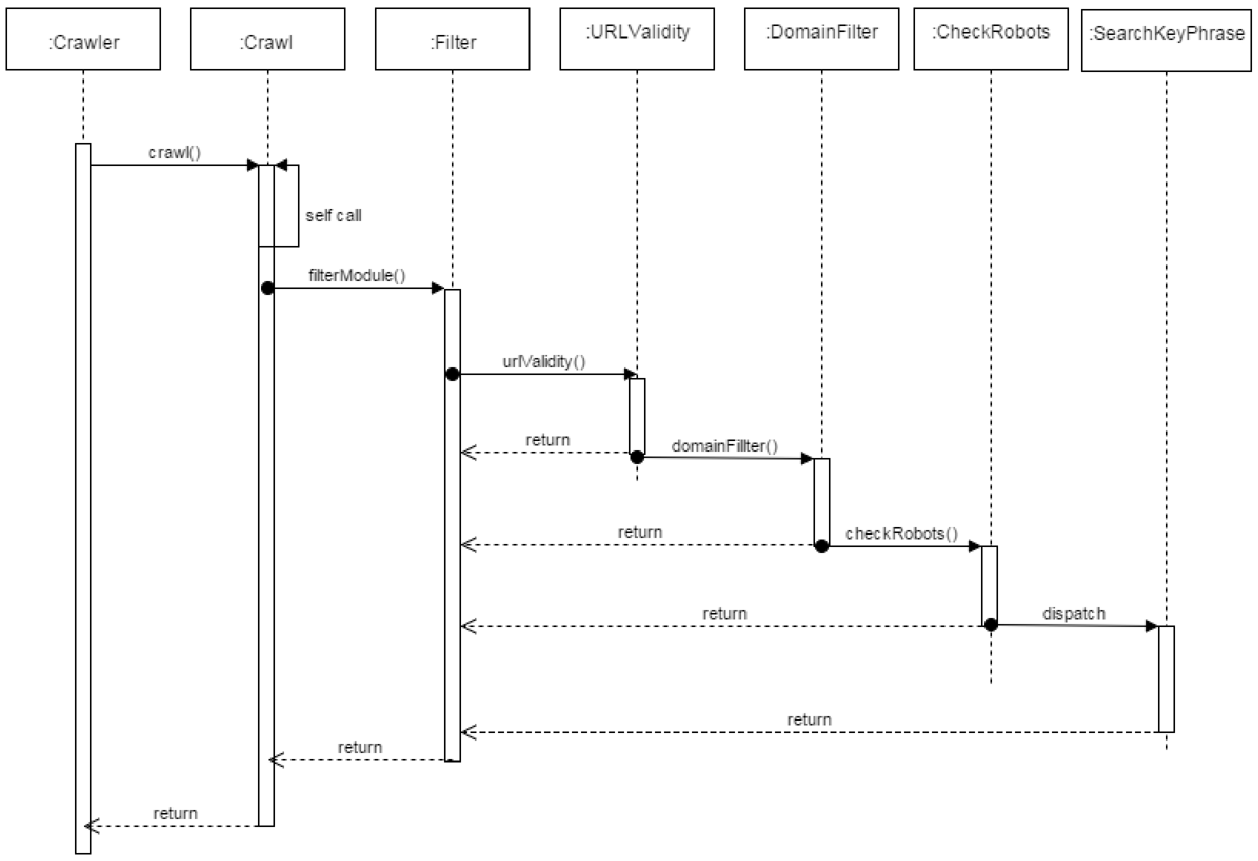
4. **checkRobots:** This module is evoked when the URL is valid. This module examines the '/robots.txt' of the corresponding webserver to determine if the URL is crawlable (i.e. allowed by the server to crawl )

5. **searchKeyPhrase:** This module does the final and core task of determining the relevance of a webpage. After '/robots.txt' gives a green signal, this module searches the whole contents of the webpage to check if the keyphrase is present.

## CONCEPTUAL MODELLING/ARCHITECTURE:

As discussed above, this crawler's architecture is modular where each module serves a function. To better understand the architecture of this crawler a UML Sequence Diagram and UML Activity Diagram is presented below.

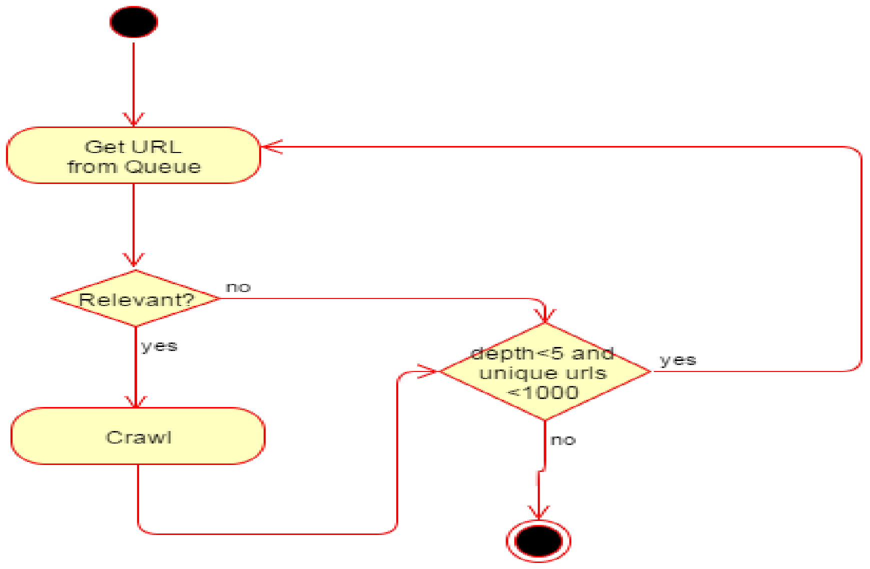
## UML SEQUENCE DIAGRAM:



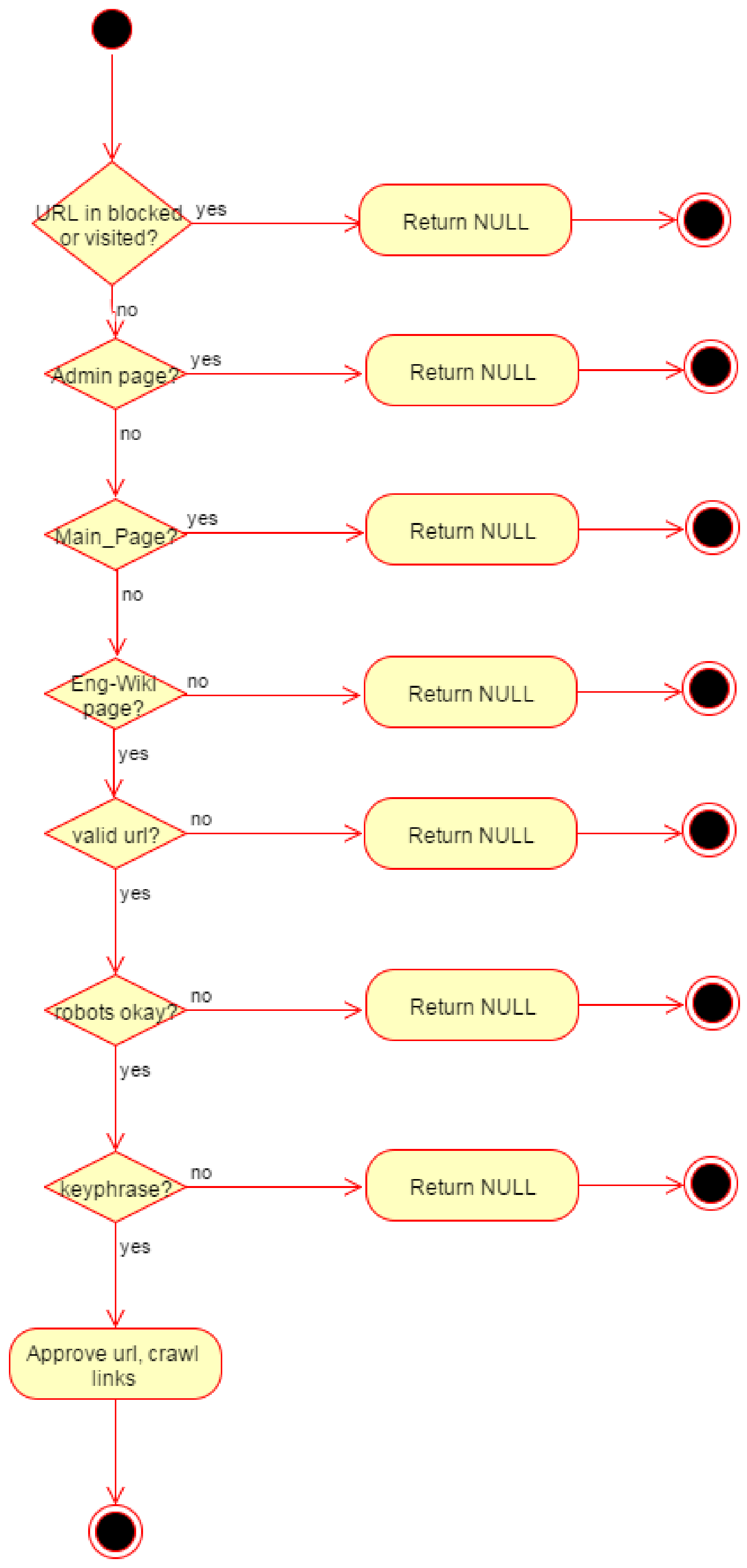
## UML ACTIVITY DIAGRAM:

This section depicts the active state transitions for the crawler as a whole and for the filter module.

## CRAWLER:



**FILTER MODULE:**



**RESULTS:**

The relevant URLs for searching without a keyphrase and searching for the word “concordance” are contained in the files URL_List_NoKeyPhrase.txt and URL_List_Concordance.txt respectively. These files are attached with this document.

**RESPONSE/CALCULATION:**

| | | |
|---|---|---|
| TOTAL PAGES RETRIEVED WITHOUT A KEYPHRASE | : | 82 |
| TOTAL PAGES RETRIEVED FOR “CONCORDANCE” | : | 13 |
| TOTAL PAGES | : | 95 |

TOTAL PROPORTION OF PAGES RETRIEVED FOR “CONCORDANCE” : 13/95

**LIMITATIONS/PROBLEM:**

An important problem that is encountered during crawling is the presence of lot of irresponsive pages. These irresponsive pages take infinite time to respond, which forces the programmer to use a global timeout on the sockets. Because of these irresponsive pages, many potential URLs are lost (couldn’t be searched or crawled). This significantly reduced the number of relevant pages listed in the results.